DJS—183 计 标 机

程序设计基础

三〇一　　罗昌隆

西 北 电 讯 工 程 学 院
无 锡 电 子 计 算 机 厂 翻 印
一九七八年七月

目　　录

第一章 计算机中数的表示及逻辑运算

§1 进位制

一、二进位制的特点

我们在日常生活中所用的数都是十进位制数。所谓数是由一个或几个数字组成，它表示一个数目。例如35是一个数，3和5都是数字，由于十进位制数是逢十进一，则35所表示的不仅仅是3和5，而是

$$3\times10^{1}+5\times10^{0}=30\times5$$

这说明当一个数字处在不同的位置时，它代表的值就不一样。十进位制数的个位数字，例如数35中的最后边的5，只能代表5不能代表其他值。但是左边的3即十位上的3就必须乘上10^{1}才能表示出它的真正的值。这个10就是十进位制的基数。

什么是基数？在十进位制系统中，10就是基数。所有的十进位制数都是由十个基本数字用不同的排列组合方法构成的。这十个基本数字是0，1，2，3，4，5，6，7，8，9。一种记数系统中所具有的基本数字个数就是基数。

所有的十进位制数都可以用从0到9这十个基本数字和以10为基数的形式表示。例如我们可将数

7390，　3425

分别写成

$$7\times10^{3}+3\times10^{2}+9\times10^{1}+0\times10^{0}$$

和

$$3\times10^{1}+4\times10^{0}+2\times10^{-1}+5\times10^{-2}$$

因为十进位制记数系统是我们熟悉的记数系统，所以使用起来感到很方便。但是在电子数字计算机上一般不采用这种进位制。因为如果采用十进位制，机器就要有十种状态来表示0，1，2，3，4，5，

6、7、8、9这十种符号。这将给机器构造带来困难。计算机一般采用二进位制。

二进位制的基数是2，只有0和1两个基本数字。因为计算机中的数是用构成计算机的元件的物理状态来表示的。我们可以用一种物理状态表示“0”，用另一种物理状态表示“1”，因此，只要有两个稳定状态的元件就能表示一位二进位制数。例如固体电路高电位和低电位，磁心的两种不同磁化方向都可用来表示1和0。

二进位制的算术运算也很简单。它和十进位制的算术运算基本上是一样的，只是它是逢二进一而不是逢十进一。例如

$0+0=0$，$1+0=1$，$0+1=1$，$1+1=10$

$0\times0=0$，$0\times1=0$，$1\times0=0$，$1\times1=1$

所有二进位制数都可用0和1两个基本数字及以2为基数的幂来表示，例如

$$10\,10\,10\,11.\,011=1\times2^{7}+0\times2^{6}+1\times2^{5}+0\times2^{4}+1\times2^{3}+0\times2^{2}+1\times2^{1}+1\times2^{0}+0\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$

$$101110=1\times2^{5}+0\times2^{4}+1\times2^{3}+1\times2^{2}+1\times2^{1}+0\times2^{0}$$

二、二进位制和十进位制间的转换

如果要将十进位制数化为二进位制数，就须用基数2连续去除。第一次除得的余数就是二进位制数的最低位数字。有余数时，余数应为1，无余数时，余数取0。然后用2除第一次所得的商，余数就是二进位制数低位上倒数第二位数字。再用2除第二次所得的商，继续上面的操作，直到不能除为止。最后一个小于2的商就是二进位制数的最高位数字。例如将46(10)转换为二进位制数，其转换步骤如下：

余数

2 | 46 ····· 0

2 | 23 ····· 1

2 | 11 ····· 1

2 | 5 ····· 1

2 | 2 ····· 0

1

$46_{(10)}=101110_{(2)}$

式中圆括号中的数表示进位制的基数。也就是说。它指出它前边的数是哪一种进位制数。

将十进位制小数转换为二进位制数的方法就是连续用2乘十进位制小数。第一次乘积中的整数部分就是二进位制小数点后第一位。把乘积的小数部分再乘以2，所得乘积的整数部分是小数点后第二位。这样，一直乘到小数部分为零或达到要求的位数。兹举例如下：

将十进位制小数0.78125转换成二进位制数

	0	$\cdot 78120_2$
第一位二进位制小数←	1	$\cdot 56250_2$
第二位二进位制小数←	1	$\cdot 12500_2$
第三位二进位制小数←	0	$\cdot 25000_2$
第四位二进位制小数←	0	$\cdot 50000_2$
第五位二进位制小数←	1	$\cdot 00000$

得 $0.78125_{(10)}=0.11001_{(2)}$

如果一个十进位制数即有整数又有小数，只须对其整数部分和小数部分，分别求出其对应的二进位制数，然后合并一起就行了。例如

$$46_{(10)}=101110_{(2)}$$

$$0.78125_{(10)}=0.11001_{(2)}$$

$$46.78125_{(10)}=101110.11001_{(2)}$$

三。八进位制

虽然二进位制有许多优点，但不是人们习惯用的计数制。一个用二进位制表示的数，所占用的位数远较相应的十进位制数为多。这样，书写时感到不便也容易出错。为了书写方便，我们在编写程序时，一般不用二进位制而用八进位。

由于8是2的三次方即

$$2^3=8$$

把二进位制数分别划成以三位一组，就能将二进位制数相应地化为八进位制数。对二进位制整数来说，要从最右边一位起向左数每三位为一组，不足三位时补上“0”。例如，对二进位整数

$$10111011$$

来说，从最右边一位起向左数每三位为一组，则得

$$10111011$$

101对应的八进位制数字为5，110对应的为6，111对应的为7。因此，

$$10111011_{(2)}=567_{(8)}$$

对二进位制小数来说，应从小数点后第一位起向右数，每三位为一组，最后不足位数补上“0”，就能得出相应的八进位制小数。例如对二进位制小数

$$0.00111101$$

来说，从小数点后第一位起向右每三位为一组，则得

$$0.001\ 111\ 010\ [0]$$ （方框内是补上的0）

注意小数点后第七位以后没有数字，凑不到三位。因此要在第七位后边加上两个0凑足三位。这一点不能忽略。001对应的八进位制数字为1，110对应的为6，100对应的为4。因此

$$0.0011101_{(2)}=0.164_{(8)}$$

八进位制数是逢八进一。它的基数为8，共有八个基数字：

0，1，2，3，4，5，6，7。

兹将若干八进位制数和它的对应二进位制及十进位制数列表对照如下：

八进位制数	二进位制数	十进位制数
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
10	1000	8
11	1001	9
12	1010	10
13	1011	11
14	1100	12
15	1101	13

从上表可以看出，一位八进位制数可表示三位二进位制数。例如二进位制整数

110001011111

共有12位。如将该数从最右边一位起每三位作一组，则可分为四组

110　001　011　111

这四组对应于八进位制的6137。这样书写的位数减少八位。所以，我们在编写程序时，一般不用二进位制而用八进位制。

八进位制数和十进位制数间的转换办法，基本上同二进位制。

§2　正数和负数的表示方法

在电子数字计算机上，参加运算的数，有的是正数有的是负数。这些数在机器中有三种不同的表示形式，即原码、补码和反码。

一、原码

原码的表示方法和普通数学所用的方法是一致的。这就是将数的正负符号放在数本身绝对值的前面。普通数学用符号“＋”或者没有任何符号表示正数，用符号“－”表示负数。

例如

＋524，　23416

等都是正数，而

－524，－23416

等都是负数。对二进位制数来说，通常规定用“0”表示正数，“1”表示负数。这个“0”或“1”就是数的符号，放在数的前面。这一位叫做符号位。例如

0110110011100011，0001101011011101

都是正数。

1101100111000 11，1001101011011101

都是负数。

上例中最高位上的“0”或“1”表示数的符号。符号后面的一串数字是数本身的绝对值。这种用符号加上绝对值的办法表示数就是原码表示法。因此，概括的话，当用原码表示法时。正数的原码就是该数的原来形式。例如

$$+1011011001101 01$$

的原码就是该数的本身

$$0101101100110101$$

对应的八进制数为 055465

对于负数，数本身不变，只是在符号位上写上“1”。例如

$$-1011011001101 01$$

的原码为

$$1011011001101 01$$

对应的八进制数为 155465

二、补码

计算机进行运算时，完全可以用数的原码进行。不过用原码进行加减时，有较大的缺点。首先机器要判断两数的符号是否相同，相同则加，不同则减。如果应做减法，又必须比较两数何者为大，然后以大减小。最后还要确定和数或差数的符号。这样，将使机器的结构复杂又降低运算速度。为了克服这些缺陷，可采用补码表示法。

为了说明什么是补码，我们先举二个例 。钟表盘上的小时数只有十二个即：1，2，3，4，5，6，7，8，9，10，11，12。超过十二小时，就又从1开始计时。如果钟停了，时钟指的是六点钟。但是，当时的正确时间是三点钟。上钟时，可将时钟倒拨三个小时。这样，我们用的是减法，即

$$6-3=3$$

$$1-7$$

时钟就能指向3。假如我们不倒拨时针，而全前即顺时针方向拨针，就必须再向前多拨九个小时，时针方能指向3。这样，我们用的是加法，即

$$6+9=15$$

从形式上看，15和3有很大差距。但是，钟表盘上的时数最大值是12，超过12就向前进位。钟表盘上不记这个进位，因此，可以把12看成零，然后从零开始向前计时。这样，13在钟表盘上就是1，14就是2，15就是3。记为

$$13=1(\mathrm{mod}12)$$

$$14=2(\mathrm{mod}12)$$

$$15=3(\mathrm{mod}12)$$

用这种方法可将减法变成加法所加的数9就是－3对于12的补码。这个9是用下式

$$12-3=9$$

计算出来的。

DJS－183机中，数规定用整数表示，每个数用16位二进制位表示。最高位（15位）为符号位。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

因此整数的值不能大于或等于$2^{16}=1000000000000000_{(2)}$，则计算负整数的补码时，就用$1000000000000000_{(2)}$减该负整数的绝对值并在符号位上写“1”。例如有负整数

-0001011010010011，它的补码就是

$$1000000000000000-0001011010010011=$$

$=1110100101101101$再将“1”写在符号位（即数的最高位）上，则1110100101101101；如果将符号也作数来处理，则求

该负整数的补码方法就是$2^{16}=1000000000000000_{(2)}$减去该负整数的绝对值。

如 $1000000000000000-0001011010010011$

$=1111010010101101$

上面谈的都是用补码表示负数的方法。对正数来说。不存在求补码的问题。它的补码就是该正数本身。例如$+0001011010010011$及$+1011011000001001$的补码仍然是0001011010010011及0101101100001001。再举一个用补码进行运算的例 如下：

例1：X=$+1011011000001001$。

Y=-0001011010010011。

求X+Y的值。

我们用补码运算。X是正数。它的补码就是它的原码。Y是负数。它的补码为1111010010101101。X+Y就归结为X与Y两个数的补码相加如下：

$$\begin{array}{r} 0101101100001001 \\ +1111010010101101 \\ \hline 10100111110110110 \end{array}$$

因为机器中符号位左边没有数位，机器将失掉最高位的“1”，得结果为0100111110110110。用补码运算的结果仍然是补码。但在这个例子中运算结果的符号位是零。这表示计算结果仍然是正数。正数的补码就是原码。因此0100111110110110就是所要的结果。

三、反码

使用补码运算可简化机器的操作。但必须事先对负数求出补码。这项求补码的操作也不够方便。但在实践中我们发现，求一个负数的补码，可以容易地先求它的反码，再求得它的补码。

一个数的反码：对正数来说，它的反码就是原码。如果是负数，求它的反码时，只需将符号位写“1”，并将其他位上的“0”换成“1”，“1”换成“0”。例如－1010011100011101及－0100101111100001的反码是1010110001100010及1101101000011110。

如果将反码和补码彼此进行比较，我们很容易看到它们之间的关系。这个关系是：反码的最低位加1就变成补码。例如

X＝－1010011100011101它的反码是1010110001100010。

如果在该反码的最低位上加1

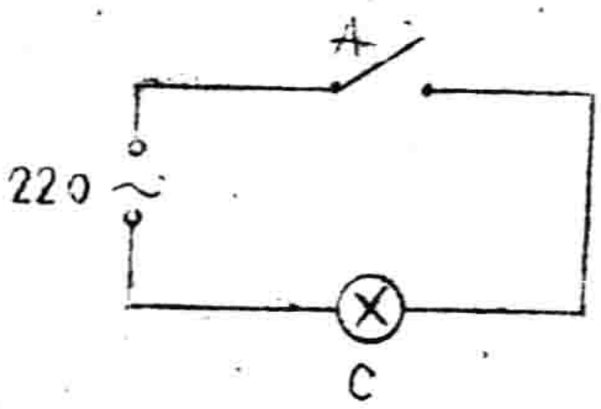
$$\begin{array}{r} 1010110001100010 \\ +0000000000000001 \\ \hline 1010110001100011 \end{array}$$

则得1010110001100011。这就是X的补码。因此，在求补码时，可利用这一关系，先求反码然后在最低位上加1。

§3 逻辑代数的概念

在人们的某些社会实践中，经常需要研究和描述某些事物之间的数量关系或者逻辑推理关系，这种逻辑推理关系用一般的语言文字来描述当然也是可以的，只是这样做太不方便，而且还难以看出它的本质方面。因此，人们在实践中总结出一种方法，借用普通数学中的语言来描述某些简单的逻辑推理关系，从而发展成一门学科，称为逻辑代数。在逻辑代数中，所研究、所探讨的量只有“0”和“1”两个量，因此，它只能用于反映一个事物的正、反两个方面。例如下面的开关电路

开关A的接通和断开。电灯C的亮和不亮。这些都是一个事物矛盾着的两个对立面。在逻辑代数中我们就可以用“1”和“0”两个量来分别表示开关A的接通和断开



$$A = \begin{cases} 0 & \text{开关A断开} \\ 1 & \text{开关A接通} \end{cases}$$

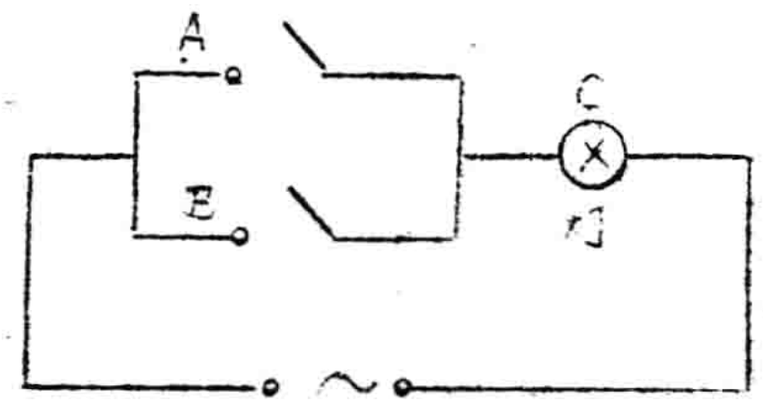
电灯C的亮和不亮

$$C = \begin{cases} 1 & \text{灯亮} \\ 0 & \text{灯不亮} \end{cases}$$

请注意。这里的“1”和“0”不是普通数学中表示数量概念的1和0，而是表示有矛盾关系的双方。上述这些事物之间的相互关系也不是普通代数中的数量关系，而是一种逻辑推理关系。这种关系可以用“1”和“0”两个量以及逻辑代数中所特别定义的一些运算规则来表示。下面我们就来介绍逻辑代数中所规定的几种基本运算。

一、逻辑加法（“或”运算）

在开关线路中并联的开关元件，对于它们之间的关系怎样来表示呢？例如为了使用上的方便，我们用两只开关控制同一只电灯：当某一开关“或”另一开关合上时，灯都发亮，当两个开关都合上时灯也发亮。这样两只开关必然是按并联方式连接的，我们称这两只开关之间的联系为“或”的关系。线路如下：



如果将开关A、B的“接通”状态记作“1”，A、B的“断开”状态记作“0”；而把灯亮记作“1”，灯灭记作“0”；则可用下式表示该线路的不同工作状态：

$$A \vee B = C$$

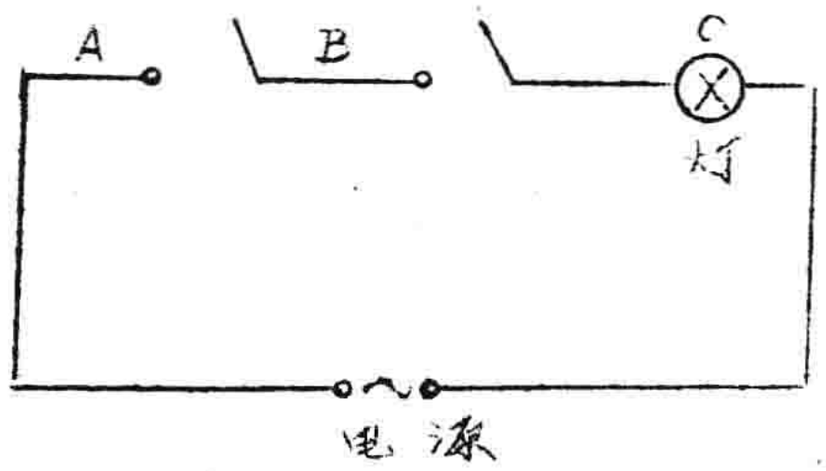
我们用“∨”表示逻辑加法（即“或”运算）符号，“A∨B”表示逻辑加法。为了书写方便，经常用“＋”代替“∨”。对于任意两个逻辑变量A、B，逻辑加法运算规则如下表所示：

A	B	C＝A＋B
0	0	0＋0＝0
0	1	0＋1＝1
1	0	1＋0＝1
1	1	1＋1＝1

这里必须指出的是，逻辑加法的运算规律与算术加法的运算规律是不相同的，要特别注意：1＋1＝1。

二、逻辑乘法（“与”运算）

对于开关线路中串联的开关之间的关系，我们怎样来表示呢？例如房间里的电灯是受房子里的开关和楼道里的总闸双重控制的，只当这两个串联的开关都合上时灯才会亮。如图



我们称这两只串联着的开关之间的联系为“与”的关系。因为只有当开关A“与”开关B同时合上，灯才会亮。如果按前例的假定条

件。则这个线路可以用下式表达：

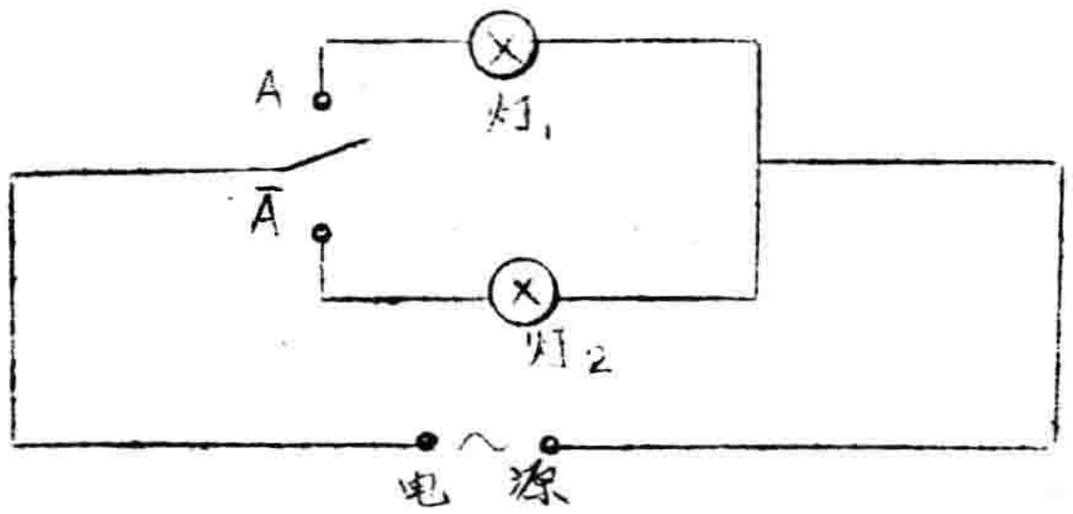
$$A \triangle B = C$$

上式表达的运算称为逻辑乘法运算（即“与”运算），为了书写方便，我们“A△B”简写为“A·B”或者“AB”。对于任意两个逻辑变量A，B，逻辑乘法运算规则如下：

A	B	C＝A△B
0	0	0△0＝0
0	1	0△1＝0
1	0	1△0＝0
1	1	1△1＝1

三、逻辑否定（“非”运算）

在开关线路中除了上述开关的并联和串联以外，还存在着这样的情况：例如用一个“单刀双掷”开关控制两个指示灯，如下图所示：



开关上合则灯$_1$亮，开关下合则灯$_2$亮。两灯的状态相反。对于这样的开关的两个触点，我们分别记作A和$\bar{A}$。当A被接通时，$\bar{A}$必须断开，而当A断开时，$\bar{A}$必然接通。因此两者是“相反”的关系。

如果用“1”表示开关的接通，用“0”表示开关的断开，则有：

若A＝1，则$\bar{A}$＝0

若A＝0，则$\bar{A}$＝1

我们称$\bar{A}$为对A进行“非”运算，$\bar{A}$读作“A反”（或“A非”）。

因此，对于任意一个逻辑变量A，其“非”运算规则如下：

A	$\overline{A}$
0	1
1	0

第二章　程序设计的一般概念

§1　人工解题的形式描述

现代的数字计算机是一种能自动地进行高速运算的计算工具，它每秒钟能进行成千上万次各种不同的运算，有的可达每秒几百万至上千万次运算。但不管它的速度多高，结构多复杂，它所进行数字运算的工作原理仍然只能是模拟人们手工计算的过程。伟大领袖毛主席教导我们："我们是马克思主义者，马克思主义叫我们看问题不要从抽象的定义出发，而要从客观存在的事实出发，从分析这些事实中找出方针、政策、办法来"。为了要了解数字电子计算机解题的工作原理，我们就应遵照毛主席的这一教导，首先从分析具体的人们最有体会的手工计算过程着手，并从中归纳出手工计算的一般规律，藉此来认识数字电子计算机解题的基本原理。

例如有一个人用算盘来计算下面题目：

$$8456+1245-3243$$

如果我们把寄存十进制数的算盘面记为R，则其计算过程的先后顺序就可列表如下：

序号	操作的命令	注　解
0	$0 \Rightarrow R$	清除盘面
1	$8456 \Rightarrow R$	在算盘中拨上8456
2	$(R)+1245 \Rightarrow R$	算盘中加上1245
3	$(R)-3245 \Rightarrow R$	算盘中减去3245
4	(R)抄送纸上	抄送运算结果
5	停止	计算结束

在执行上述5步操作之后，在算盘R中有如下运算的结果：